

1 Predicting real-time traffic conflicts using deep learning

2
3 **Nicolette Formosa, Corresponding author**

4 School of Architecture, Building and Civil Engineering,
5 Loughborough University
6 Loughborough LE11 3TU United Kingdom
7 E-mail: N.Formosa@lboro.ac.uk
8

9 **Mohammed Quddus**

10 School of Architecture, Building and Civil Engineering,
11 Loughborough University
12 Loughborough LE11 3TU United Kingdom
13 Tel: +44(0)1509 228545
14 E-mail: M.A.Quddus@lboro.ac.uk
15

16 **Stephen Ison**

17 Department of Politics, People and Place,
18 Leicester Castle Business School,
19 The Gateway, Leicester, LE1 9BH, United Kingdom
20 Tel: +44(0)116 257 7647
21 E-mail: Stephen.ison@dmu.ac.uk
22

23 **Mohamed Abdel-Aty**

24 Department of Civil, Environmental and Construction Engineering,
25 University of Central Florida
26 Orlando, FL 32816, USA
27 Tel: +1(407)823-4535
28 E-mail: M.Aty@ucf.edu
29

30 **Jinghui Yuan**

31 Department of Civil, Environmental and Construction Engineering,
32 University of Central Florida
33 Orlando, FL 32816, USA
34 E-mail: jinghuiyuan@knights.ucf.edu
35

Abstract

Recently, technologies for predicting traffic conflicts in real-time have been gaining momentum due to their proactive nature of application and the growing implementation of ADAS technology in intelligent vehicles. In ADAS, machine learning classifiers are utilised to predict potential traffic conflicts by analysing data from in-vehicle sensors. In most cases, a condition is classified as a traffic conflict when a safety surrogate (e.g. time-to-collision, TTC) crosses a pre-defined threshold. This approach, however, largely ignores other factors that influence traffic conflicts such as speed variance, traffic density, speed and weather conditions. Considering all these factors in detecting traffic conflicts is rather complex as it requires an integration and mining of heterodox data, the unavailability of traffic conflicts and conflict prediction models capable of extracting meaningful and accurate information in a timely manner. In addition, the model has to effectively handle large imbalanced data. To overcome these limitations, this paper presents a centralised digital architecture and employs a Deep Learning methodology to predict traffic conflicts. Highly disaggregated traffic data and in-vehicle sensors data from an instrumented vehicle are collected from a section of the UK M1 motorway to build the model. Traffic conflicts are identified by a Regional–Convolution Neural Network (R-CNN) model which detects lane markings and tracks vehicles from images captured by a single front-facing camera. This data is then integrated with traffic variables and calculated safety surrogate measures (SSMs) via a centralised digital architecture to develop a series of Deep Neural Network (DNN) models to predict these traffic conflicts. The results indicate that TTC, as expected, varies by speed, weather and traffic density and the best DNN model provides an accuracy of 94% making it reliable to employ in ADAS technology as proactive safety management strategies. Furthermore, by exchanging this traffic conflict awareness data, connected vehicles (CVs) can mitigate the risk of traffic collisions.

Keywords: Safety Surrogate Measures, traffic conflicts, data integration architecture, Regional–Convolution Neural Network (R-CNN), Deep Neural Network (DNN)

1. Introduction

Recent breakthroughs in technology, digital infrastructure, machine learning and big data computing will transform the way we will plan, undertake, interact, make decisions and use our built environment, vehicles and transport infrastructure for the movement of people and goods. Intelligent vehicles, enabled by connectivity and automation, cloud computing, artificial intelligence and Internet of Things (IoT), will allow unprecedented capability to collect, exchange and analyse large volumes of heterodox data for optimal decision making in real-time and will therefore increasingly revolutionise our economy and society over the coming decades.

For instance, ADAS (advanced driver assistance system) technologies have the potential to transform the automotive sector (estimated revenue of \$89.3 billion in 2025 according to Tractica, 2016) as they protect occupants, drivers and reduce collisions through monitoring, warning, braking, and steering tasks. Since driver error contributes to over 90% of traffic collisions (NHTSA, 2015), the underpinning principle of ADAS is to support the driver in driving safely including reducing human involvement from the loop of driving as necessary. Fuelled largely by regulatory and consumer interest in safety applications (for instance, both the European Union and the United States are mandating that all vehicles be equipped with autonomous emergency-braking systems and forward-collision warning systems by 2020), a recent McKinsey survey suggests that consumers are becoming even more interested in ADAS applications that promote comfort and economy, such as those that assist with parking or monitoring blind spots (Mohr et al., 2013). ADAS utilises different vehicle-based safety surrogate measures (SSMs) in order to warn drivers of any potential conflict. This is largely based on a pre-specified threshold of a single metric – *time-to-collision* (TTC) or its modified version rather than other important SSMs such as speed variance, yaw rate, lateral and

1 longitudinal deceleration/acceleration rates, time gap and distance-to-collision along with their
2 variant by traffic density, speed and weather.

3 In addition, the automotive industry is currently designing and manufacturing
4 Autonomous Vehicles (AVs). However, at present, the safety aspect of these automated driving
5 has primarily been tackled with data and information gathered from in-vehicle sensors and
6 equipment. Given the fact that a sensor failure is inevitable even with redundant measurements
7 in a chain of operations, a holistic approach is required to guarantee a collision-free path
8 planning of automated driving (Katrakazas et al., 2017). For example, a conflict detection
9 technique may provide a critical hint of any collision-risk arising from the contextual and
10 circumstantial factors such as collision hot-spots identified through historical collision data,
11 unsafe traffic dynamics in real-time, infrastructure deficiencies, complex road layout, and other
12 imminent threats from surrounding objects. A real-time traffic conflict detection technique will
13 also be very important and useful for forming effective platoons in connected vehicles (CVs).

14 It is, therefore, fundamental to develop a real-time conflict detection method suitable for
15 ADAS, CVs and AVs. The overarching complexity in building such a model lies in the
16 management, integration and use of off-line and real-time data which are large and heterodox in
17 many ways such as uncertainty, sampling frequency, completeness, structure, consistency and
18 accuracy. In addition, an appropriate modelling technique, which can handle this large and
19 complex data, is required. Earlier modelling techniques have focused on nonparametric methods
20 and machine learning models, such as a regression tree model (Hossain and Muromachi, 2013),
21 support vector machine (Yu and Abdel-Aty, 2013) and genetic programming (Xu et al., 2013a).
22 Nonetheless, these techniques cannot perform the analysis required for big data with complex
23 characteristics and have not addressed the issues emerging when the data is highly imbalanced.

Therefore, the results emerging from such techniques are inclined to be profoundly influenced by the large classes (e.g. ‘safe’ traffic dynamics) while neglecting the smaller ones (e.g. traffic conflicts). This is because they aim to minimise the overall error rate rather than paying specific attention to the smaller classes. On the other hand, Deep Learning (DL) in particular Deep Neural Network (DNN) models with a Mean Squared False Error loss function have the ability to capture classification errors from both majority and minority classes equally (Wang et al., 2016). They are also able to elicit complex and non-linear patterns to classify and predict while catering for big data (Najafabadi et al., 2015). Therefore, DL methodology would be an ideal candidate in addressing the complexities discussed earlier.

The purpose of this paper is two-fold: (1) to design a data integration architecture for heterodox data and (2) to develop a real-time traffic conflict detection method using DL methodology. A section of the M1 motorway in the UK was used as a case study from which data was collected using an instrumented vehicle. Due to the large complex datasets collected from multiple sensors, a data integration architecture was developed to extract meaningful and accurate information in a timely manner. From the video data, the safe traffic dynamics and traffic conflicts were identified and together with the other collected, and estimated factors (SSMs and traffic variables), the DNN model was developed. Parts of the data were utilised to validate the performance of models using K-fold cross-validation and to determine the performance and ability of the model in predicting link-level traffic conflicts.

2. Literature Review

Timely detection of traffic conflicts is fundamental to proactively mitigate the risk of collision through ADAS, CVs and AVs. This has been very challenging as multiple factors act concurrently and the methods employed to detect conflicts have some severe weaknesses. This

review is therefore structured into two main parts: factors affecting conflicts and methods to detect them.

2.1 Factors affecting real-time conflicts

SSMs are used as an indirect measure of traffic safety as they can be effectively utilized to proactively identify safe traffic dynamics and traffic conflicts. By understanding how different measures interact with each other in developing a traffic conflict, evasive manoeuvres are recommended by proactive safety systems such as ADAS. In fact, a traffic conflict is identified when the value of a SSM exceeds a critical threshold (Zaki et al., 2013).

There are different property based SSMs adopted to estimate the likelihood that an unsafe condition arises. In the case of estimating risk behaviours, the time-based SSMs have been adopted since they unify both spatial proximity and speed difference (Laureshyn et al., 2010). However, the time-based SSMs are not always sensitive for spatial proximity, when the maximum braking capacity is applied. In this case, distance-based SSMs have found to be appropriate since the risk of a collision is relatively high if the required distance for executing an evasive action is less than the available distance. On the other hand, when an evasive manoeuvre in a traffic conflict is detected, such as braking, SSMs based on deceleration are important to assess the kinematic characteristics involving risk avoidance. As a result, each property-based conflict indicators represent partial images of the true severity of critical traffic events. Typical SSMs based on three properties as identified in the literature are presented in Table 1. These SSMs are primarily related to car-following and lane-changing related traffic conflicts. For each of the SSMs, the definition, critical value and additional comments are presented.

Table 1

SSM factors

SSM	Definition	Critical Value	Comments
Post-Encroachment Time (PET) Time-based	The time taken for the rear bumper of the preceding vehicle to leave the zone of a common spatial zone till the front bumper of the second vehicle enters the same zone (Songchitruksa and Tarko, 2006).	A PET value less than 1 or 1.5 seconds is considered as critical (Archer, 2005).	This measure while reflecting the drivers' behavior, does not consider the impact of the collision. $PET \approx \frac{R}{V_{FV}}$
Time To Collision (TTC) Time-based	The remaining time before an impact takes place between two road users unless an event to change their present physical parameters occurs.	The critical value for TTC is given as 1.5 seconds.	An assumption is that the consecutive vehicles have to sustain constant speeds. $TTC = \frac{R-L}{V_F - V_L} \quad \forall V_{FV} > V_{PV}$
Modified TTC (MTTC) Time-based	This SSM proposed by (Ozbay et al., 2008) follows the TTC indicator but considers combinations of relative speeds and accelerations in the various car following scenarios.	A 4 second threshold value for the MTTC measure was proposed in (Ozbay et al., 2008).	The MTTC value by itself does not provide enough indication about the seriousness of the collision. $MTTC = \frac{-\Delta V \pm \sqrt{(\Delta V)^2 + 2 \Delta a \Delta X}}{\Delta a}$
Proportion of Stopping Distance (PSD) Distance-based	The ratio between the remaining distance (RD) based on the assumption that the maximum deceleration rate is used to the potential point of collision and the Minimum acceptable Stopping Distance (MSD).	A PSD value less than 1, is regarded as 'unsafe' and collisions cannot be avoided if the MADR is considered	PSD is regarded as a good surrogate indicator and has been used for safety evaluation by (Guido and Saccomanno, 2010). $PSD = \frac{RD}{MSD} \quad MSD = \frac{V_{FV}^2}{2a_{FV}^{min}}$
Deceleration Rate to Avoid a Crash (DRAC) Deceleration-based	The rate at which the following vehicle must decelerate to come to a timely stop to prevent a collision with other conflicting vehicles or to obtain the same speed of preceding vehicle to avoid a rear-end collision (Xiaobo Qu et al., 2014)	A DRAC value of more than 3.35m/s ² is considered as being in a traffic conflict (Archer, 2005).	DRAC gives a better measure to represent both frequency and severity of high-risk vehicle interactions. Many researchers have acknowledged the relevance of DRAC as a safety-performance measure, e.g.(Archer, 2005; Guido and Saccomanno, 2010). $DRAC = \begin{cases} \frac{(V_{FV} - V_{PV})^2}{\Delta X}, & \text{if } V_{FV} > V_{PV} \\ 0, & \text{otherwise} \end{cases}$
Criticality Index Function (CIF) Deceleration-based	This aim of this index is to consider the degree of danger and the probability of an imminent conflict even if a collision does not essentially occur (Chan, 2006).	Same as TTC	This index can be used as a surrogate safety measure to evaluate the probability and the severity of a collision. $CIF = \frac{V_{FV}^2}{TTC}$

In Table 1, R is the range of the two vehicles, L is the length of preceding vehicle, ΔX is $(R - L)$, V_{FV} and V_{PV} denote the speed of the following and preceding vehicle respectively, $MADR$ is the maximum available deceleration rate and a_{FV}^{\min} is typically set as 3.92m/s^2 .

Nevertheless, traffic variables also play an important role in predicting crash occurrences, and the quantification of their impact has been utilised to formulate countermeasures in improving traffic safety (Wang et al., 2015). This is because unsafe road occurrences take place when traffic moves, and it is, therefore, natural to explore traffic characteristics to understand their impact on dangerous conditions. In fact, many studies have been carried out to analyse the connection between a potential crash probability and various traffic parameters (Xu et al., 2013b). A few patterns have been observed in the literature about how each variable affects unsafe conditions, despite that mixed evidence is presented. These traffic variables are inter-related and are usually classified as traffic speed, traffic flow, density and occupancy (Wang et al., 2013). Moreover, recent research indicates that SSMs are heavily influenced by traffic variables. For instance, Papazikou et al. state that critical TTC values are different at different operating speeds of the roadway (Papazikou et al., 2017).

Therefore, by embedding these traffic variables in conjunction with the SSMs as factors in proactive safety systems to predict traffic conflicts may be beneficial. This adds more information about the network and the environment through which the ego-vehicle is driving. The inherent challenge in developing such a safety automotive system is that intelligent vehicles are equipped with a growing number of complex components which need to run simultaneously while also being time-critical because of the safety requirements (Zheng et al., 2016). These components include the sensors used such as radar, camera and GPS to estimate the SSMs and to collect real-time traffic variables. This data also needs to be processed using computational units

such as processors and GPUs. These sensors and computational units are then connected through various buses while spanning different languages, interfaces and hardware and processing a high volume of data to perceive the environment accurately and efficiently. Therefore, such systems need to adopt a valid and reliable data integration architecture system in order to process large and heterodox data in real-time.

2.2 Modelling techniques for real-time conflict detection

A fundamental challenge when it comes to predicting rare events in the real world is the unequal distribution between its classes creating a highly imbalanced dataset (He and Garcia, 2009). In this study, traffic conflicts are also observed less frequently than safe traffic dynamics. This can compromise the performance of some machine learning predictors since they are based on the assumption of balanced class distributions. To overcome this issue, studies identified in the literature have made use of case-control designs when developing prediction models (e.g. Abdel-Aty et al., 2005). However, since some of the data has been omitted, it may influence the calibration of the model resulting in an inaccurate output. To obtain a more realistic and reliable prediction from the machine learning predictor, which genuinely reflects the underlying classification a full dataset should, therefore, be adopted (Yang et al., 2018).

Existing studies have mainly used shallow prediction models, which cannot be applied to the real-world complex highly imbalanced dataset. On the other hand, in work by Sameen and Pradhan (2017), a DL approach is adopted, and their research reveals the capability of this methodology to overcome the data imbalance problem. Yang et al. (2018) also address this imbalance problem by using a DL approach. Again, their results achieve a powerful identification ability in real-time prediction. Hence since the data used in this study is complex, large, heterodox and highly imbalanced, a DL methodology is adopted. Additionally, this study

1 takes advantage of its superior performance and the powerful ability to tackle high-dimensional
2 data.

3 In summary, it has been revealed that there is a considerable amount of work undertaken
4 in the identification of the factors which contribute to traffic conflicts. However, these studies at
5 best provide mixed conclusions about their efficiency and applicability. This highlights the need
6 for the use of more than one factor so as to get a clear understanding of the conflict mechanism
7 (Laureshyn et al., 2010). Therefore, by adding both the traffic variables with the SSMs as factors
8 to develop a model to predict traffic conflicts this increases the system's reliability and efficiency
9 (Nadimi et al., 2016). However, this is not a simple task as it requires a reliable centralised data
10 architecture system to allow for the data from sensors to be processed, synchronised and
11 embedded as factors in the model. Nevertheless, this data is highly complex and imbalanced, and
12 a DL methodology has to be adopted to cater for this data.

13 **3. Development of a Data Integration Architecture and Data Collection**

14 Real-time vehicle kinematics and traffic conditions data have been collected using an
15 instrumented vehicle. However, a large amount of data cannot be processed using conventional
16 systems because of many forms of data, their complexity and their varying sampling frequencies.
17 Hence, to examine the information and patterns within the data, a system architecture is needed
18 for data collection, transmission and storage. In order to unify and process all the information on
19 a central computational unit, a centralised data integration architecture was developed to satisfy
20 the requirements of various sensor environments and provide a run-time environment. The
21 advantage of this architecture is that the combination and arrangement of the system allow all the
22 functions to share information more conveniently. Nevertheless, the centralised architecture
23 requires high computation capability. The architecture transmits two kinds of information:

continuous information of the vehicles localisation and velocity, and information about the surrounding dynamic environment. It is developed as shown in Figure 1:

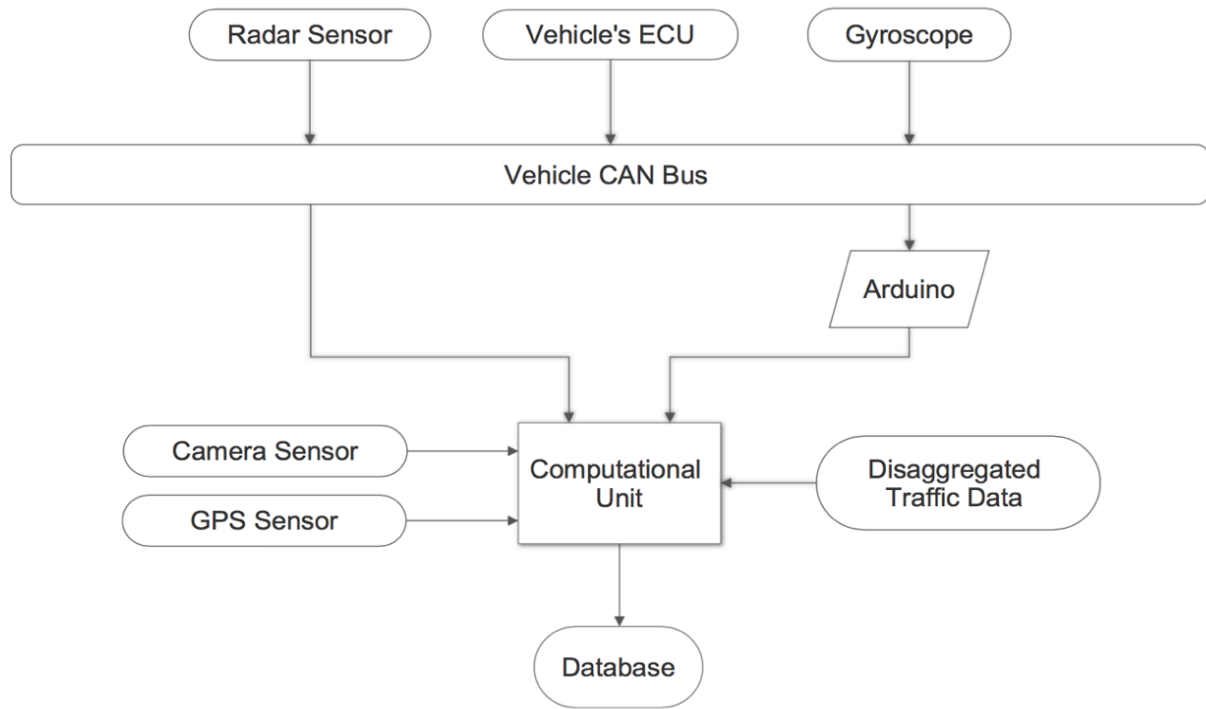


Fig. 1. Centralised data integration architecture

In this study, the vehicle's speed and the yaw rate are collected by the Electronic Control Unit (ECU), and the gyroscope respectively by scanning the Controller Area Network (CAN) bus at a rate of 15Hz. To increase the accuracy and quality of the data, these messages are then linked to the radar sensor messages. The vehicle speed computed from the radar sensor alone is not sufficient. This is because, if the ego-vehicle is travelling at the same speed as one of the detected vehicles the latter will appear static. However, by noting the speed of the ego-vehicle, the actual speed of the detected vehicle can be computed. These combined messages are then read via an Arduino microcontroller and transferred via a USB to a computational unit where all the other sensors (Radar, Camera and GPS) and disaggregated traffic data are connected to the same computational unit.

The instrumented vehicle is installed with a radar sensor (ARS 308-21) and a Grasshopper3 camera (GS3-U3-41C6C-C). The localisation sensor installed is a U-Blox EVK-M8 GNSS system with DR which enables simple evaluation of the high-performance positioning technology. The disaggregate traffic flow data is obtained from Motorway Incident Detection and Automatic Signalling (MIDAS). Each loop detector records traffic information such as vehicle speed, occupancy, flow and headway. The information is defined per minute by lane and collected from the 88 loop detectors found in the study area with a spacing of nominally 400m. The data is obtained from the loop detector with the lowest Euclidean distance between the ego-vehicle and the loop detector. A summary of the measures from sensors and loop detectors are presented in Table 2.

Some of the issues associated with data collection relate to: missing data, communication problems and spurious data due to hardware or software problems. Another issue for loop detector data is the ‘no-flow’ data when no vehicles appeared within the current minute (O’Reilly et al., 2004). Some errors can be observed from unrealistic values for example:

- (i) speed values < 0 km/hr,
- (ii) occupancy $< 0\%$ or $> 100\%$,
- (iii) flow = 0 veh/min and speed > 0 km/hr,
- (iv) flow = 0 veh/min, speed = 0 km/hr and occupancy $> 0\%$,
- (v) flow > 0 veh/min and speed = 0 km/hr ,
- (vi) flow > 0 veh/min, speed > 0 km/hr and occupancy = 0%.

All the observations from these sensors were gathered in their respective program, are then extracted from the sensing system and stored in a database. The data from each sensor is

- 1 synchronised creating a *centralised* integrated architecture in which all data, measurement
- 2 readings of multiple variables is gathered in one place rather than on multiple computation units.

3 Table 2

4 List of Measures

	Variables based on Source	Description and Use	Units
Data from sensors within the ego-vehicle	GPS (Frequency of 1Hz)		
	Position – Ego (X,Y)	Longitude and Latitude which specify the precise location of the ego vehicle on the surface of the Earth.	°/min
	Speed - Ego	The rate of change of position with time.	m/s
	Vehicle's ECU and Gyroscope messages sent through the CAN Bus and read using an Arduino (Frequency of 15Hz)		
	Velocity – Ego	The rate of change of position with time.	m/s
	Acceleration - Ego	The rate of change of velocity with time.	m/s ²
	Yaw Rate - Ego	The rate of change of angular velocity.	°/s
	Radar (Frequency of 15Hz with a range of 200m and azimuth angle of 56°)		
	Longitudinal Displacement	Range in the x-direction between the target and the ego-vehicle.	m
	Lateral Displacement	Range in the y-direction. Determines lane classification.	m
	Relative Longitudinal Speed	Relative speed between the ego and the target vehicle; from which the speed of the target vehicle can be calculated.	m/s
	Relative Lateral Velocity	The component of the velocity of the target vehicle which is resolved in the y-direction.	m/s
	Relative Lateral Acceleration	Relative acceleration between the ego and the target vehicle; from which the acceleration of the target vehicle can be calculated.	m/s ²
	Camera (Frequency of 15Hz)		
	Video data	Record surrounding road traffic environment at an image resolution of 2048x2048	fps
External data to the ego-vehicle	Loop Detectors (per lane) (Frequency of 0.0167Hz i.e. 1-min interval)		
	Speed	The average speed for each lane	km/hr
	Flow	The total number of vehicles passing through each lane	vehicles/min
	Occupancy	The percentage of time the monitoring site space was occupied by a vehicle for each lane	0-100%
	Headway	The difference between the time when the front of a vehicle arrives at a point and the time the front of the next vehicle arrives at the same point	s

The data used in this study was collected from a section of the UK M1 motorway between junctions 17 and 23. The total length of this corridor is about 55km with 3-4 lanes on each side and ten slip roads. A total of 15 trips were conducted between April 2017 and December 2018 at different times between 10am and 7pm from multiple drivers. This is because it is known that drivers behave differently under diverse traffic and environmental conditions. As a result, data was also collected in various traffic conditions such as highly congested and free flow conditions as well as in different weather conditions to cover most of the events that could occur in the real-world. The total duration of these trips adds up to approximately 19 hours of data with 1,200,000 observations at 15Hz. Since the instrumented vehicle has only a forward-facing radar and camera, we can only study traffic conflicts associated with car-following and lane-changing.

4. Methodology

Developing a conflict detection model is challenging as data on traffic conflicts and their influencing factors are required. However, traffic conflicts data are not available and therefore, a method is developed to identify traffic conflicts using data from an instrumented vehicle (see Figure 1). The conflict identification process from the collected data consists of training an R-CNN network to identify unsafe conditions by further detecting vehicle and lane geometry data acquired through image processing techniques from a single front-facing camera. This data is later integrated with the data from other sensors and fed into a DNN model to be trained in order to predict traffic conflicts on the road. The methodology of this study consists of two main components: (1) conflict identification and (2) conflict detection. A functional block diagram of the methodology undertaken for conflict identification and classifier development is presented in Figure 2.

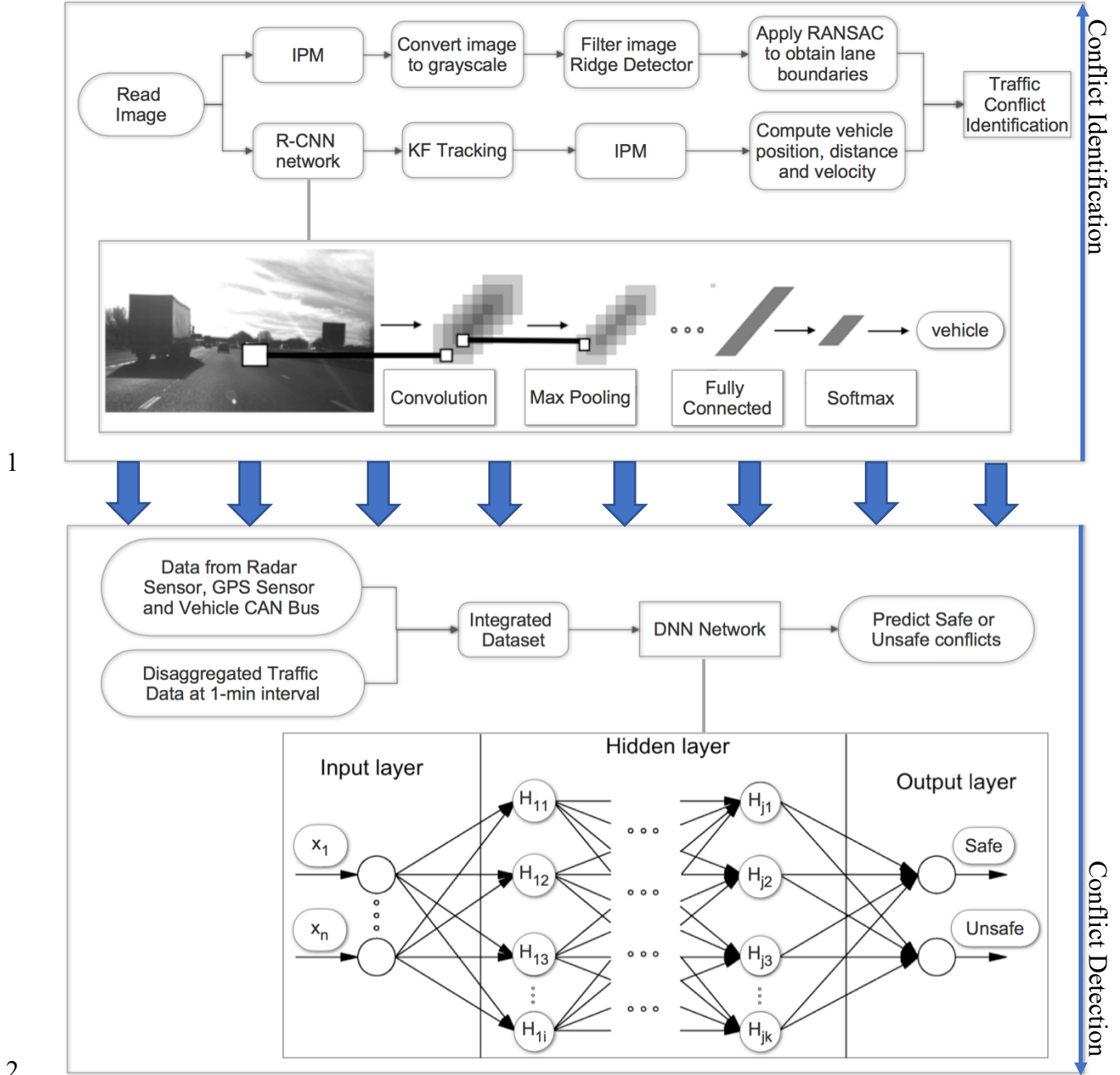


Fig. 2. The functional block diagram of the methodology undertaken

4.1 Conflict event identification

A new method is developed to identify traffic conflicts based on the data collected from our instrumented vehicle. In order to identify traffic conflicts on the road, preceding vehicles were detected, tracked and classified accordingly. Initially the video data is segregated into a large

number of images. Ground truth labels on these images to train and test the model is necessary for the model to be successful. However, obtaining high quality labelled data is costly and time-consuming (Yamashita et al., 2018). A CNN learning classifier is trained to detect vehicles using this ground truth data set, by combining bottom-up regional proposals with rich features (R-CNN) (Zhao et al., 2018).

The building blocks of an R-CNN model includes convolution layers, pooling layers and fully connected layers. A typical architecture is made up of repetition of two types of layers: convolutional layers and pooling layers, followed by one or more fully connected layers. The role of the convolutional layer is to carry out feature extraction by applying a kernel across the input made from an array of numbers usually termed as tensor. Each element of the kernel is multiplied to every input tensor and summed to obtain an output value organised in feature maps. This procedure is repeated by applying multiple kernels to form a number of feature maps. These feature maps represent different characteristics of the input tensors. Two key hyperparameters that define the convolution operation are kernel size and the number of kernels. These are typically chosen based on training dataset. An important note is that in order to fit the centre of the kernel at the outermost element and to retain the same in-plane dimensions zero padding is added to this layer. Moreover, the distance between two successive kernel positions (stride) is often of value 1. The results are then passed through a non-linearity activation function such as a rectified linear unit (ReLU), identified as the half-wave rectifier $f(z) = \max(z, 0)$.

A pooling layer offers a process which reduces the in-plane dimensionality of the feature maps in order to introduce a translation invariance to small shifts and distortions and decrease the number of successive learnable parameters (Yamashita et al., 2018). In this particular layer, there are no learnable parameters, whereas filter size, stride, and padding are hyperparameters in

pooling operations, similar to convolution operations. Max pooling is the pooling operation adopted in this research. It extracts patches from the input feature maps and uses only the maximum value in each patch, while deleting the other values. A commonly used max pooling layer is with a filter of size 2×2 with a stride of 2. This decreases the sample size of the in-plane dimension of feature maps by a factor of 2. While this affects the height and width, the depth dimension of feature maps is retained.

The output feature maps of the final convolution or pooling layer is usually flattened in which every input is connected to every output by a weight. The fully connected layers are the fundamental building blocks of the R-CNN model and, the composition of each layer creates a highly non-linear system. Every layer is trained using stochastic gradient descent (SGD), consisting in presenting the input vector with examples, estimating the outputs and the errors, computing the average gradient for those examples, and adjusting the weights accordingly. The process is iterated until the average of the objective function stops decreasing. It is described as stochastic since each set of examples provided a noisy estimate of the average gradient over all examples. This procedure usually finds a good set of weights, so the network is able to extract features and learn the different sizes and shapes of the vehicles. Every fully connected layer is also followed by a nonlinear function. The final layer outputs bounding boxes indicating the regions occupied by a vehicle and the corresponding confidence.

A summary of the list of parameters (optimised during training) and hyperparameters (set beforehand) in an R-CNN with the values used in this study is given in Table 3.

Table 3

List of parameters and hyperparameters in an R-CNN

R-CNN	Parameters	Hyperparameters with values
Convolution layer	Kernels	<ul style="list-style-type: none"> • Kernel size (3x3) • No. of kernels (random) • Stride (1) • Padding (0) • Activating function (ReLU)
Pooling layer		<ul style="list-style-type: none"> • Filter Size (2x2) • Stride (2) • Padding (0)
Fully connected layer	Weights	<ul style="list-style-type: none"> • No. of weights • Activation function (ReLU)
Training process		<ul style="list-style-type: none"> • Model architecture (15 layers) • Optimizer • Learning Rate (1e-4) • Loss function (SGD) • Mini-batch size (128) • Epochs (60) • Regularisation • Weight initialisation (random)

Since the R-CNN model is not pixel accurate and the position and size of bounding box changes, the results from this model can be noisy. To filter this noise, the position and dimensions of bounding boxes are tracked with a Kalman Filter (KF) having a Constant Velocity state transition model (Li and Jilkov, 2003). The KF also has the ability to predict how these measurements will change in time by making use of a sequence of measurements and a mathematical model. Hence, it also provides the prediction of the position and size of the bounding boxes which increases the certainty within the results. This is because these can then

1 be compared with the actual visual in the next time step. The KF procedure is given in two stages
2 as follows:

3 Prediction Stage:

$$4 \quad \hat{x}_{t|t-1} = A\hat{x}_{t-1} \quad (1)$$

$$5 \quad P_{t|t-1} = AP_tA' + Q \quad (2)$$

6 Correction Stage:

$$7 \quad K_t = P_{t|t-1}C'(CP_{t|t-1}C' + R)^{-1} \quad (3)$$

$$8 \quad \hat{x}_t = \hat{x}_{t|t-1} + K_t(y_t - C\hat{x}_{t|t-1}) \quad (4)$$

$$9 \quad P_t = (I - K_tC) P_{t|t-1} \quad (5)$$

10 where A is the constant velocity state transition model, Q is the process noise covariance, R is
11 the measurement noise covariance, z is the measurement vector consisting of the position and
12 size of the bounding box and x is the state vector consisting of the position and size of the
13 bounding box and their rates of change, K_t is the optimal Kalman gain that minimises the
14 residual error, I is an identity matrix, P represents the error covariance matrix and C gives the
15 observation model.

16 To associate data from the R-CNN model and KF, the multi-object tracker accepts
17 detections from multiple sensors and assigns them to tracks using a global nearest neighbour
18 criterion. To ensure that the R-CNN and KF are assigned to the same track, the normalised
19 distance of the detections from the track must be less than a defined threshold. In the case of
20 discontinuous data, the tracks are deleted, and the procedure has to be reinitiated. These
21 discontinuities in the data need to be known beforehand as otherwise it will affect the generated

1 traffic conflict output. However, it is typically known during data collection that an error has
2 occurred.

3 The detection data is stored in a matrix in which the timestamp, x-position, y-position,
4 height and width of bounding box is given for each frame at 15Hz. From this data, the distance
5 and the velocity of the detected vehicles can be estimated by using the Kalman filter predictions
6 and Inverse Perspective Transform (IPM) technique. The IPM technique is calibrated to
7 transform pixel coordinates from the image plane to the road plane. The transformation matrix
8 that maps points from the image plane to the road plane is defined by:

$$9 \text{ transformation matrix} = \left(D * \left(T * (B * E) \right) \right)^{-1} \quad (6)$$

10 where D is the intrinsic matrix, T is the translation matrix, B is the rotation matrix and E is the
11 projection matrix of the camera. An estimate of the camera's intrinsic, extrinsic and the lens
12 distortion parameters is obtained by calibrating 20 images of a checkerboard calibration pattern
13 using the same camera used to collect data. The intrinsic parameters describe the internal
14 characteristics of the camera, such as the focal length of the lens, optical centre, and lens
15 distortion coefficients. These parameters help to improve image quality, correct for lens
16 distortion and map real-world distances to pixels. On the other hand, the extrinsic parameters
17 define the camera's position (translation) and angle (rotation) in space in relation to a fixed
18 object which is essential to structure from motion.

19 Therefore, by adopting this methodology of transforming the forward-facing image to a
20 'bird's eye view' image, the perspective distortion of the road surface and the non-linearity of
21 distances is eliminated. In order to compute the distance, the coordinates at the bottom edge of
22 the bounding box (which is the region where the vehicle touches the ground) are transformed
23 from the image plane to the road plane using the IPM.

While the states of the KF include the rate of change of position and size of the bounding box, the velocity is not readily obtainable since the states cannot be converted to the closure rate in meters per second. Instead, the states defining the rate of change are used to predict the position of the bounding box after 1 millisecond. Therefore, the current vehicle position and the predicted vehicle position are converted to the road plane and are used to calculate the velocity and acceleration based on the time difference between the current position and the predicted one.

Additionally, the bird's eye view images are used to identify sudden or unexpected lane changes (LCs). In a bird's eye view, lane markers appear parallel and of uniform thickness making it easier to detect. These images are converted to grayscale and then filtered with a ridge detector. The resulting binary image indicated pixels belonging to the lane markers. A random sample consensus (RANSAC) algorithm was then used to robustly generate candidate second order polynomials that characterize the apparent lines in the binary image.

From the resulting information obtained from the images, an automated video analysis system was developed to identify potential traffic conflicts associated with LC and rear-end (RE) conflicts. These conflicts are generally identified through response to an evasive action, and the temporal and/or spatial proximity (Zheng et al., 2014). The determining factors to develop this system were based on the following criteria: time, distance, speed, braking, acceleration and deceleration.

4.1.1 Lane change conflicts

A LC conflict can result in either a sideswipe conflict or a RE conflict. If a Lane Changing Vehicle (LCV) and the EV maintain their velocity and the traffic conflict angle is between $[5^\circ, 85^\circ]$, this would result in a potential sideswipe conflict. However, when the traffic conflict angle occurs between $[0^\circ, 5^\circ]$, the LC conflict is instead considered as a RE conflict (Kusano et al.,

2014) and the criteria defined for RE conflicts are applied. Moreover, a LC conflict exists provided that the lateral acceleration of the EV is less than 0.07g. This threshold guarantees that the EV doesn't change lane when it identifies a cut-in vehicle (Yang et al., 2019) or that it not an overtaking manoeuvre by the EV (Chen et al., 2015). Therefore, a potential LC traffic conflict is identified if:

- a) the total time headway between the LCV and the EV is less than 3s, plus the time for the LCV to enter the EV's lane which can be calculated by: $t_{LC} = l_{LCV}/V_{LCV}$ (Qu *et al.*, 2014) where l_{LCV} and V_{LCV} is the length and velocity of the LCV. Since TTC is later used as one of the factors in the DNN model, time headway is instead employed to identify conflicts as TTC and time headway are considered independent of each other (Vogel, 2003).
- b) the lateral distance between LCV and EV is greater than 1.5m and less than 2.2m showing that the LCV intends to start its movement to the EV's lane (Yang et al., 2019) while the EV is accelerating longitudinally or moving with constant velocity.
- c) the longitudinal available space is less than the maximum longitudinal distance of 75m between the LCV and the EV. This is because this distance is appropriate to determine when two vehicles are interacting (Bham, 2009; Yang et al., 2019).
- d) the rate of change of speed when the LCV initiates the LC manoeuvre till it stabilises within the boundaries of the EV's lanes is greater than 20% in absolute value. This is because it implies that an unstable deceleration and/or acceleration is taking place which can suggest an adverse impact on the EV (Yang et al., 2019).

4.1.2 Rear-End Conflicts

A RE conflict takes place when the EV nearly collides with a vehicle in its forward pathway provided that the traffic angle conflict occurs between $[0^\circ, 5^\circ]$. From the video analysis system, a RE traffic conflict is highlighted if:

- a) the time headway is less than 3s when the lateral distance is less than 2m (Kusano et al., 2014) and the lateral acceleration is less than 0.07g. The lateral distance of 2m is used even though it is slightly larger than half the lane width to allow for flexibility in identifying preceding vehicles while the lateral acceleration confirms that the EV is not changing lane.
- b) the speed of the EV is greater than the speed of the PV during deceleration of the PV, and whilst the EV is accelerating or moving at a constant velocity.
- c) the speed of the PV is greater than the speed of the EV, but the PV is decelerating and the EV is accelerating or moving at constant velocity.
- d) the PV exceeds its threshold for a braking event. This is set at 0.3g which results in acceleration or deceleration of $\pm 2.943\text{m/s}^2$ respectively. If this is exceeded, a traffic conflict is identified. This threshold value is also adopted in Naito et al., (2009) and Miyajima et al., (2011).
- e) the EV brakes and its speed is greater than 10mph for at least 1s of the braking time. This ensures that it is not continuously braking due to a congested environment.

Merging and diverging areas are also considered in this study. The most common conflicts occurring in these areas are also RE and LC (Wang et al., 2017). As a result, the same criteria developed for both RE and LC conflicts are adopted when a merging or diverging section is identified.

The development of this video data analysis system accelerated the procedure and ensured that any errors associated with human observation in the first stage of the conflict identification system were eliminated. Based on these criteria, each traffic conflict identification was validated by manually observing the corresponding videos collected from the forward-facing camera. As a result, a significant amount of time was saved as the points observed were only those identified and highlighted as potential traffic conflicts by the system. During manual observations, some of the events were misclassified and not included to prevent bias in the results.

4.2 Classifier development

Following the identification of traffic conflicts, a DNN model is developed to predict traffic conflicts based on a set of influencing factors. For the development of the DNN, it requires both the output (i.e. tracked vehicles which are classified as ‘threats’ as discussed in the previous section) and the input (corresponding estimated SSMs and traffic variables) for the same timestamp. These are fed into the DNN network to train and identify patterns of what arises from the factors to result in a traffic conflict.

While the input and output layers of the network can be directly accessed, the hidden layers made up of a number of neurons, work as a black box. These layers are connected together utilising weights and biases. Indeed, different hidden layers with a different number of nodes are employed in order to structure DNN models for each activation function. Two different nonlinear activation functions are adopted throughout the network to train the model: *tanh* and *maxout*.

The *tanh* activation function is a rescaled and shifted logistic function known for its symmetry about zero, and so the training also converges faster. It is formulated as:

$$\tanh(x) = \frac{2}{1+e^{-2x}} - 1 \quad (7)$$

while the *maxout* function is a generalisation of the Rectified Linear activation in which each neuron picks the most significant output of separate channels and each channel has its own weights and bias values, is formulated as:

$$\text{maxout}(x_1, x_2) = \max(x_1, x_2) \quad (8)$$

Therefore, the nonlinear transformation is carried out by these functions. The data is transferred throughout the hidden layers and the output from each neuron p the hidden layer is obtained as:

$$H_{lp} = f_{lp} \left(\sum_{q=1}^{n_{l-1}} w_{lpq} (H_{(l-1)q}) + b_{lp} \right) \quad (9)$$

where H_{lp} refers to the output from hidden layer l and neuron p , f_{lp} is the activation function for layer l and particular neuron p , w_{lpq} is the weight coefficient for the q^{th} input for layer l and neuron p , and b_{lp} is the bias parameter for layer l and neuron p . Both the number of hidden

layers and neurons are varied to identify the optimal output y_O from the output layer O , given as:

$$y_O = f_O \left(\sum_{q=1}^{n_t} w_{Oq} (H_{Oq}) + b_O \right) \quad (10)$$

where f_O is the softmax activation function, w_{Oq} and b_O are weights and bias in the output layer and n_t is the total number of neurons in final hidden layer H_O . Nevertheless, when the data is imbalanced and a classification is performed, the cross entropy loss function is optimised by the model and is given as (Candel and Parmar, 2015):

$$L(k) = - \sum_{y \in \{0,1\}} \left(O_{\hat{y}}(k) * \ln(O_y(k)) + (1 - O_{\hat{y}}(k)) * \ln(1 - O_y(k)) \right) \quad (11)$$

where $O_y(k)$ and $O_{\hat{y}}(k)$ are the actual and predicted output respectively, from output layer O and iteration k . Since big data is used in the study, overfitting can become a problem hence regularisation techniques can be adopted. This is done either by changing the Lasso Regularisation (L1) or Ridge Regularisation (L2) in order to modify the loss function, or by

applying the dropout technique in which the network is forced to learn multiple independent representations of the data.

In order to evaluate the performance of the DNN model, K-fold cross-validation is a commonly used technique (Rodríguez et al., 2010). For instance, a 10-fold cross-validation means that the data is divided into 10-folds. The total number of iterations are equal to the number of folds, and while keeping one-fold for validation, the remaining folds are used for training. The AUC values give the estimation resulting from each fold. Moreover, the performance of the DNN model can be evaluated based on the following criteria where TP is equivalent to true positive, FN is false negative, FP is false positive, and TN is true negative:

1. Sensitivity - This variable gives the prediction accuracy of how many conditions are predicted as traffic conflicts out of all the present traffic conflicts.

$$\text{Sensitivity} = \frac{TP}{TP+FN} \quad (12)$$

2. Precision – This variable shows how many conditions are predicted as traffic conflicts, out of all the predicted traffic conflicts.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (13)$$

3. False alarm rates – This represents the number of safe conditions predicted as traffic conflicts.

$$\text{FAR} = \frac{FP}{FP+TN} \quad (14)$$

4. Accuracy – This shows if both ‘safe traffic dynamics’ and ‘traffic conflict’ cases are predicted correctly.

$$\text{Accuracy} = \frac{TP+TN}{TP+FN+FP+TN} \quad (15)$$

1 5. Area Under Curve values – This variable gives the predictive performance of the model.
2 It usually ranges between 0-1 and the higher the AUC value, the better the performance.

3 Additionally, several methods are available to construct a scoring system for variable importance
4 such that the most influencing factors which contributed to a traffic conflict detection can be
5 deduced. For the DNNs developed in this paper, the methodology developed by Gedeon (1997)
6 is applied and the basis of these scores are the network's connection weights.

7 **5. Results**

8 **5.1 Data Integration Architecture**

9 The data integration architecture developed in this study offers a system to integrate data from
10 multiple sensors in real-time in a heterogeneous environment. In fact, this architecture provides a
11 reliable storage to access and extract meaningful information from a big dataset and
12 communicate findings of several factors in a timely manner.

13 A unique feature of this analysis is that it is able to show how the different parameters estimated
14 vary with other variables. More specifically, three examples are presented of how SSMs vary
15 with traffic variables in Figure 4.

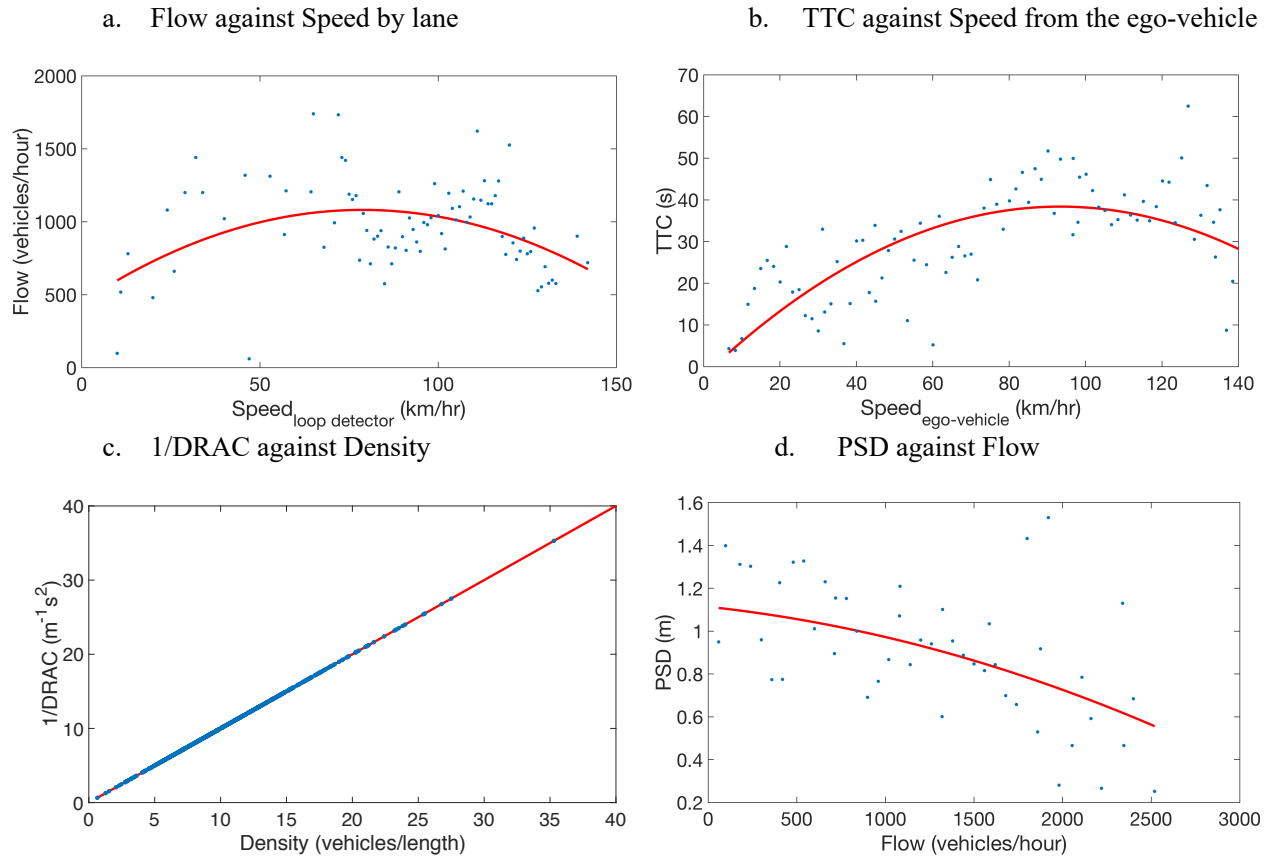
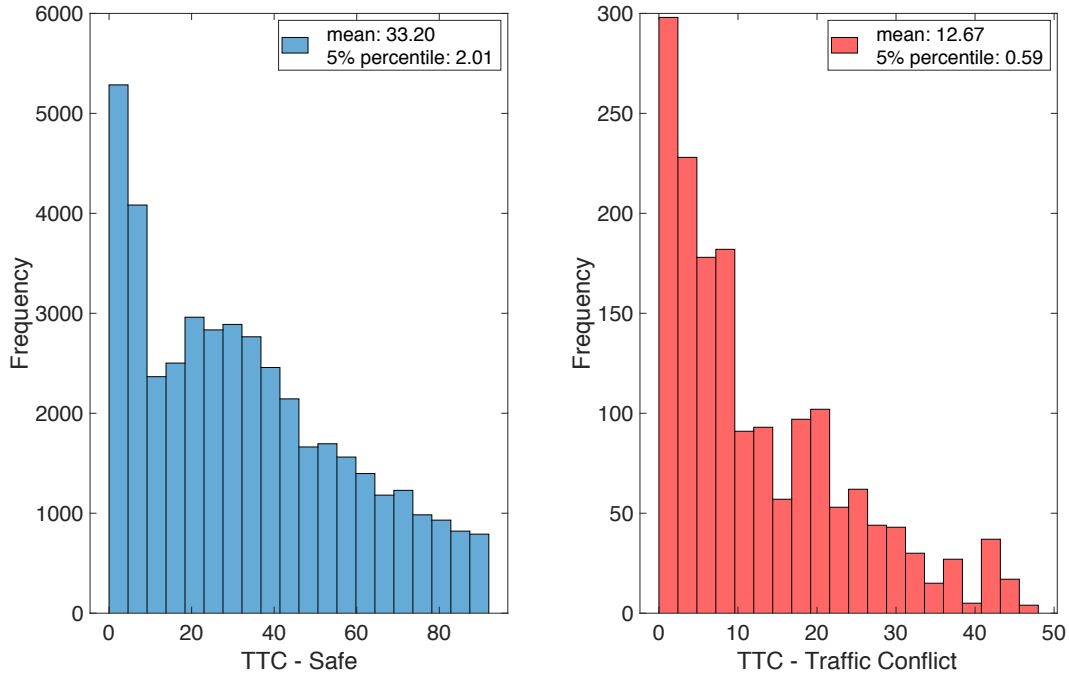


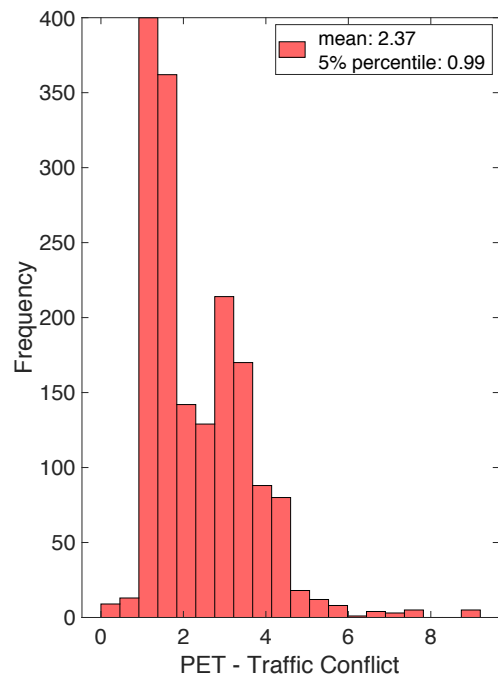
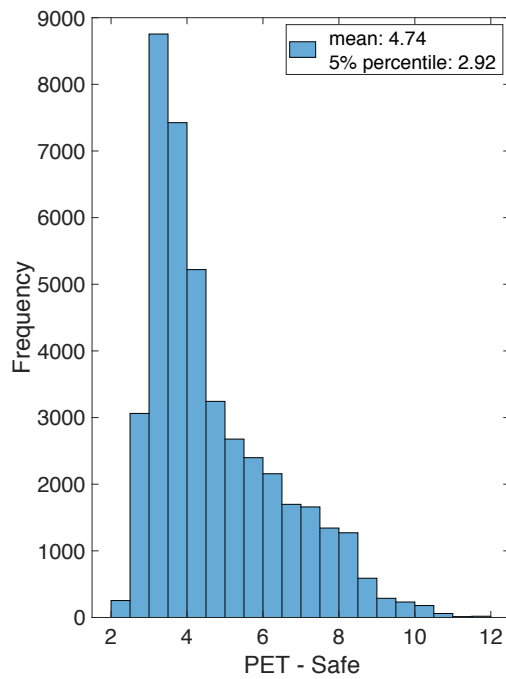
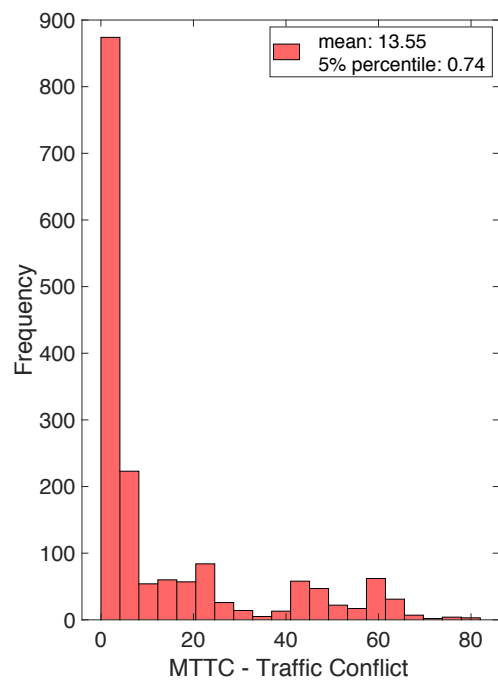
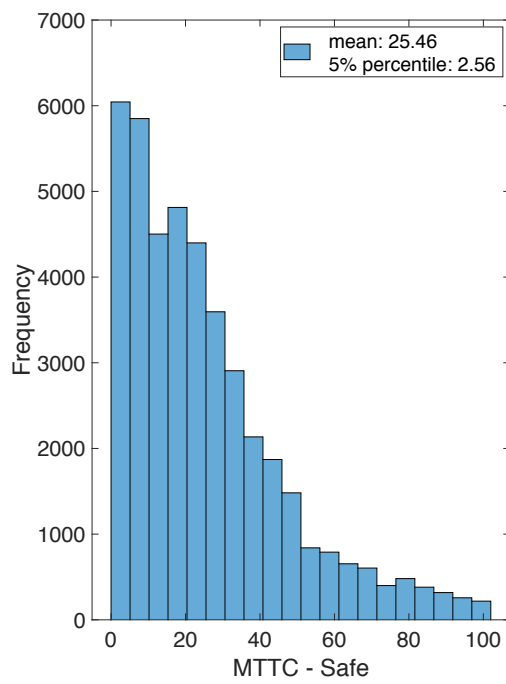
Fig. 4. The variation of factors with the same timestamp

These results clearly show that a relationship exists between traffic variables and SSMs. A different property-based SSM (time, distance and deceleration) is tested, and in each case, an association can be identified. In fact, in Figure 4(a) the general relationship is obtained between flow and the speed, Figure 4(b) shows that an inverse quadratic relationship exists between TTC and ego-vehicle speed, Figure 4(c) shows a linear relationship such that as the density increases the deceleration decreases and Figure 4(d) shows that as the flow increases hence there is less space on the road the PSD value decreases as well. This highlights the need for their effect to be embedded in the DNN model to add insight in identifying traffic conflicts and improve the predictability performance. Additionally, each SSM exhibits a varying relationship with a traffic

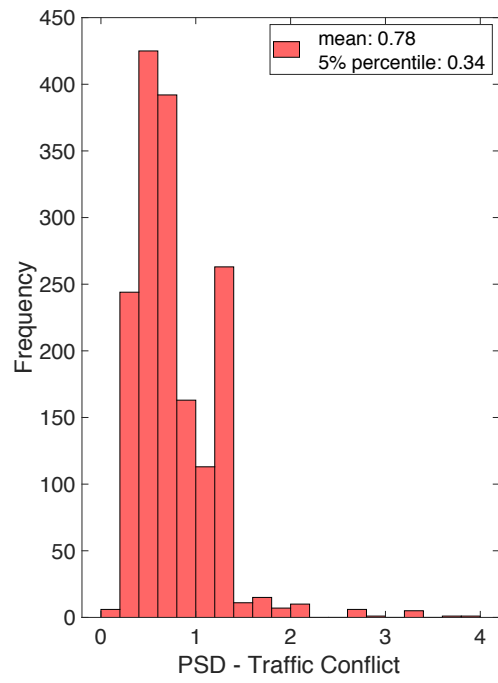
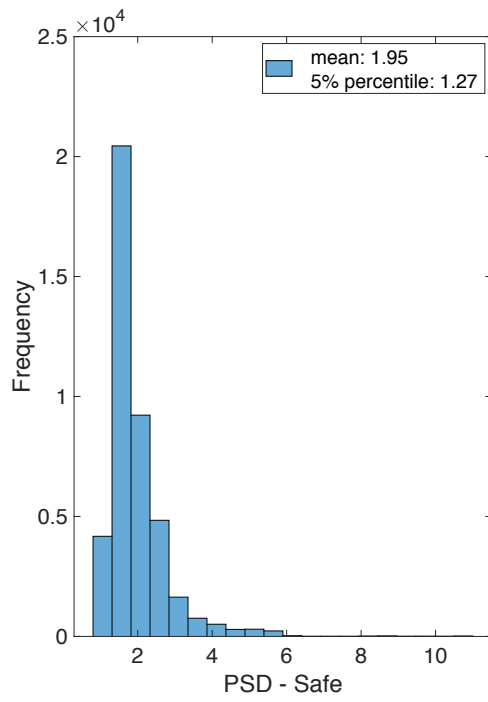
1 variable which suggests the need for adopting more than one SSM. Moreover, to test the
2 thresholds for each SSM found in the literature (see Table 1), from the data integration system,
3 the SSM distributions during safe traffic dynamics and traffic conflicts were extracted and are
4 presented in Figure 5.



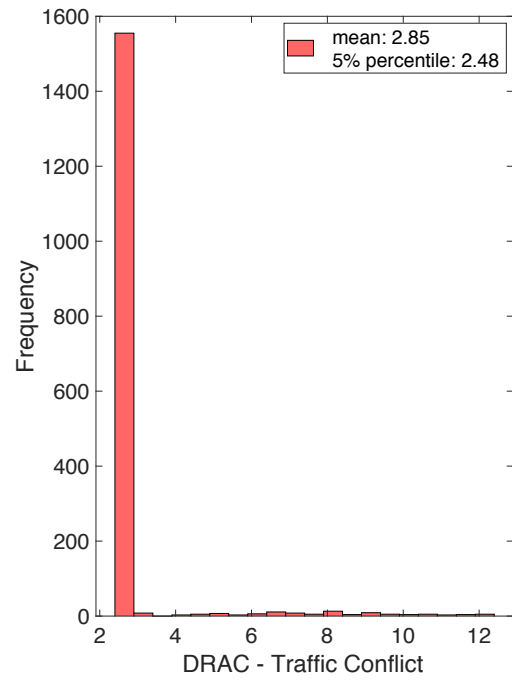
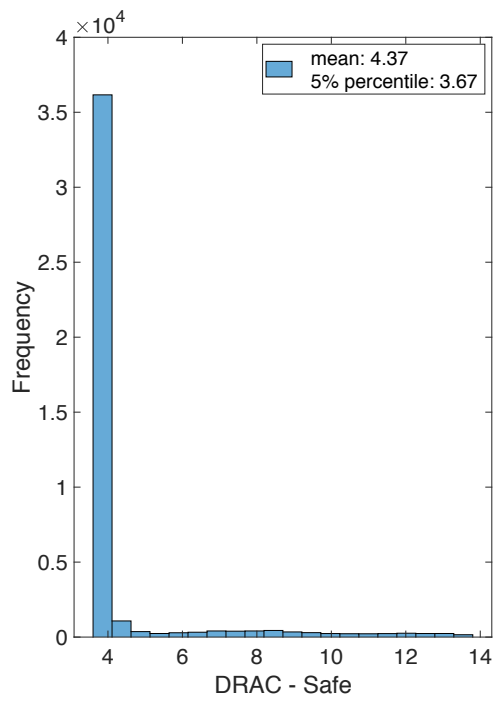
5



1

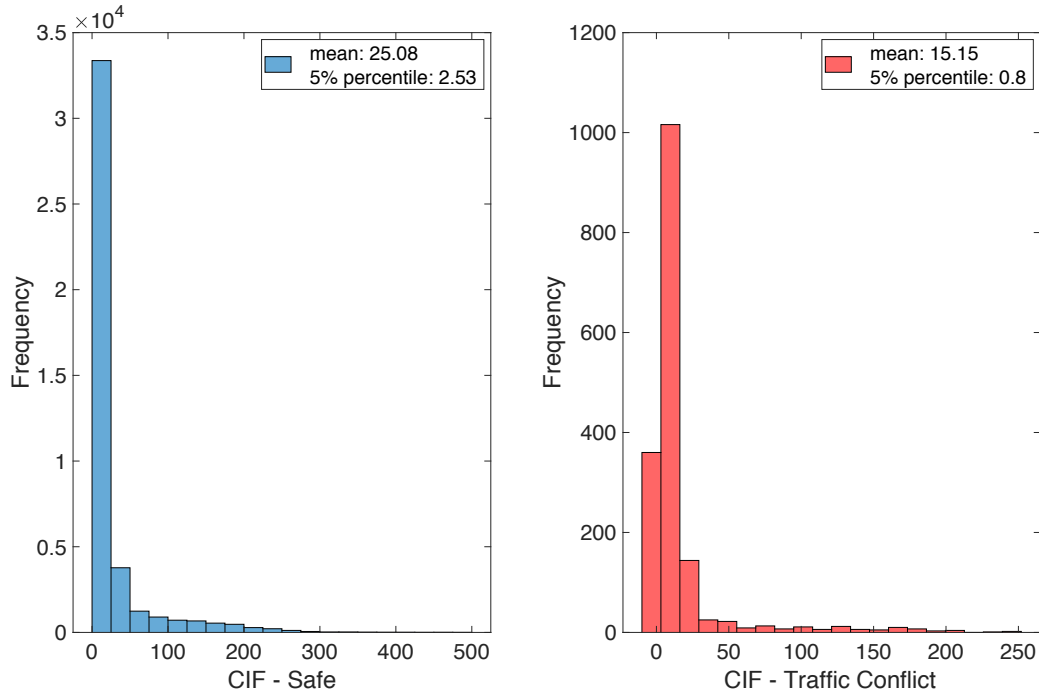


2



3

1



2

3 **Fig. 5.** The distribution for each SSM during safe and traffic conflicts

4 Figure 5 shows the distribution of each SSM together with their mean and the 5% percentile
 5 value. In particular, the threshold values for each SSM presented in Table 1 can be compared to
 6 the 5th percentile value. This is because there is more interest in the left-hand side of the
 7 distribution (Papazikou et al., 2019) since the lower it is, the more unsafe it is. As a result, the 5th
 8 percentile value is more representative than the mean value and can be compared to the values
 9 found in literature.

10 These values obtained for each SSM give a clear indication that the values are always lower in a
 11 traffic conflict condition relative to the ‘safe’ traffic conditions. This shows that their influence as
 12 factors in the DNN model are beneficial since they would be able to add insight in predicting traffic
 13 conflicts. Nevertheless, some values for the SSMs are still large despite that a traffic conflict is
 14 present. This happens in particular for TTC because of the nature of its equation. For example, the

TTC equation will result in a large TTC value even if the gap between the vehicles is very small and the change in velocity between the consecutive vehicles is also very small. This large value can be misleading at some points, however, by using multiple SSMs as factors in the model rather than just relying on one SSM, this bias is eliminated.

5.2 Evaluation of the R-CNN Model using Testing Data

To evaluate the performance of the R-CNN model to identify vehicles in the images, training was carried out on 4,200 images accounting for 70% of the data. Each image included the region of interest for vehicles. For detection purposes, the input size of the image input layer was chosen as 32x32x3 since smaller portions of the image should be analysed. A total of 15 layers were setup to develop the network in which every layer was trained in mini-batches of 128 images for 60 epochs using SGD with a learning rate of 0.0001. In the evaluation process, the detected bounding boxes that had a ratio of 0.7 overlap with ground-truth boxes, were identified as correct while boxes with no overlap were identified as misdetection. Based on 1,800 images in the testing data, the average precision of the network is 0.675. This network was used throughout for all the video data to identify vehicles and based on the criteria developed the conflicts were identified surrounding the ego-vehicle.

5.3 Real Time Traffic Conflict Detection Model

A total of 26 variables were used as input variables to the DNN model consisting of all the six SSMs presented in Table 1, the ego-vehicle's speed, speed variance between lanes, lane number, traffic variables by lane (i.e. speed, density, flow, headway, occupancy) and their mean and the standard deviations across all lanes. The input variables of the DNN model have been fine-tuned with maximum effort by considering the weighting factor between the layers during the training

process of multiple DNN models. The variables with consistently high weighting were the only used variables to develop the predictor. The output variables extracted from the video data consisted of 42,541 safe traffic dynamics and 1,663 traffic conflicts. The dataset was then randomly partitioned into 70%-30% for training and testing data respectively as it provides the best trade-off. As a result, the training data consisted of 29,785 safe conditions and 1,158 traffic conflicts while the testing data consisted of 12,756 safe conditions and 505 traffic conflicts.

The DNN model was developed by testing two of the activation functions: *tanh* and *maxout*, and by varying the number of hidden layers i.e. 2, 3, 4, 5 layers, and the number of nodes i.e. 10, 12, 15, 18, 20, 25, 30 resulting in 28 DNN modes for each activation function. The best 4 models together with their performance metrics are presented in Table 4.

Table 4

Performance metrics in the DNN model

No	Activa- tion	Hidden layers	Train	Validation	Test Data				
			AUC	AUC	Sensiti vity	False alarm Rate (FAR)	Precision	Accuracy	AUC
1	<i>Tanh</i>	[30, 30, 30, 30]	0.940	0.938 s.d(0.0301)	66.7%	3.0%	47.2%	0.959	0.935
					70.5%	5.0%	36.0%	0.941	
					75.4%	7.0%	30.0%	0.924	
					78.8%	10.0%	23.5%	0.895	
					89.7%	20.0%	14.7%	0.795	
					92.8%	30.0%	11.3%	0.709	
2	<i>Tanh</i>	[18, 18, 18, 18]	0.936	0.930 s.d(0.0099)	64.0%	3.0%	46.1%	0.958	0.922
					68.3%	5.0%	35.2%	0.940	
					68.7%	7.0%	27.9%	0.921	
					78.6%	10.0%	23.5%	0.895	
					85.9%	20.0%	14.5%	0.801	
					91.7%	30%	10.8%	0.709	
3	<i>Maxout</i>	[20, 20]	0.936	0.931 s.d(0.0186)	67.3%	3.0%	48.2%	0.960	0.927
					77.8%	5.0%	37.8%	0.943	
					79.2%	7.0%	31.2%	0.926	
					83.1%	10.0%	24.8%	0.897	
					91.5%	20.0%	15.1%	0.804	

					94.3%	30.0%	10.9%	0.709	
4	<i>Maxout</i>	[15,15]	0.939	0.931 s.d(0.0186)	63.4%	3.0%	46.0%	0.958	0.913
					72.1%	5.0%	37.6%	0.943	
					75.0%	7.0%	30.7%	0.925	
					78.6%	10.0%	24.1%	0.896	
					90.7%	20.0%	15.3%	0.806	
					92.9%	30.0%	11.0%	0.730	

1 The AUC values based on the test data presented in this Table, resulted in four high values
 2 indicating that the DNN network can provide a good prediction performance. By comparing the
 3 test data AUC values for each model, Model 1 has the best performance. This model also
 4 achieved the highest AUC for the training, validation and testing data. The AUC values for all 4
 5 models are all relatively high which shows that this methodology achieves good classification
 6 and has a strong prediction ability. In addition, based on Model 1, the most influential factor was
 7 found out to be: traffic flow followed by speed variance between the lanes while the most
 8 influential SSM was distance-based PSD. Furthermore, from Figure 6, the ROC curves for each
 9 model rise rapidly towards the upper-left-hand corner of the graph which reflects the good
 10 performance of each model tested.

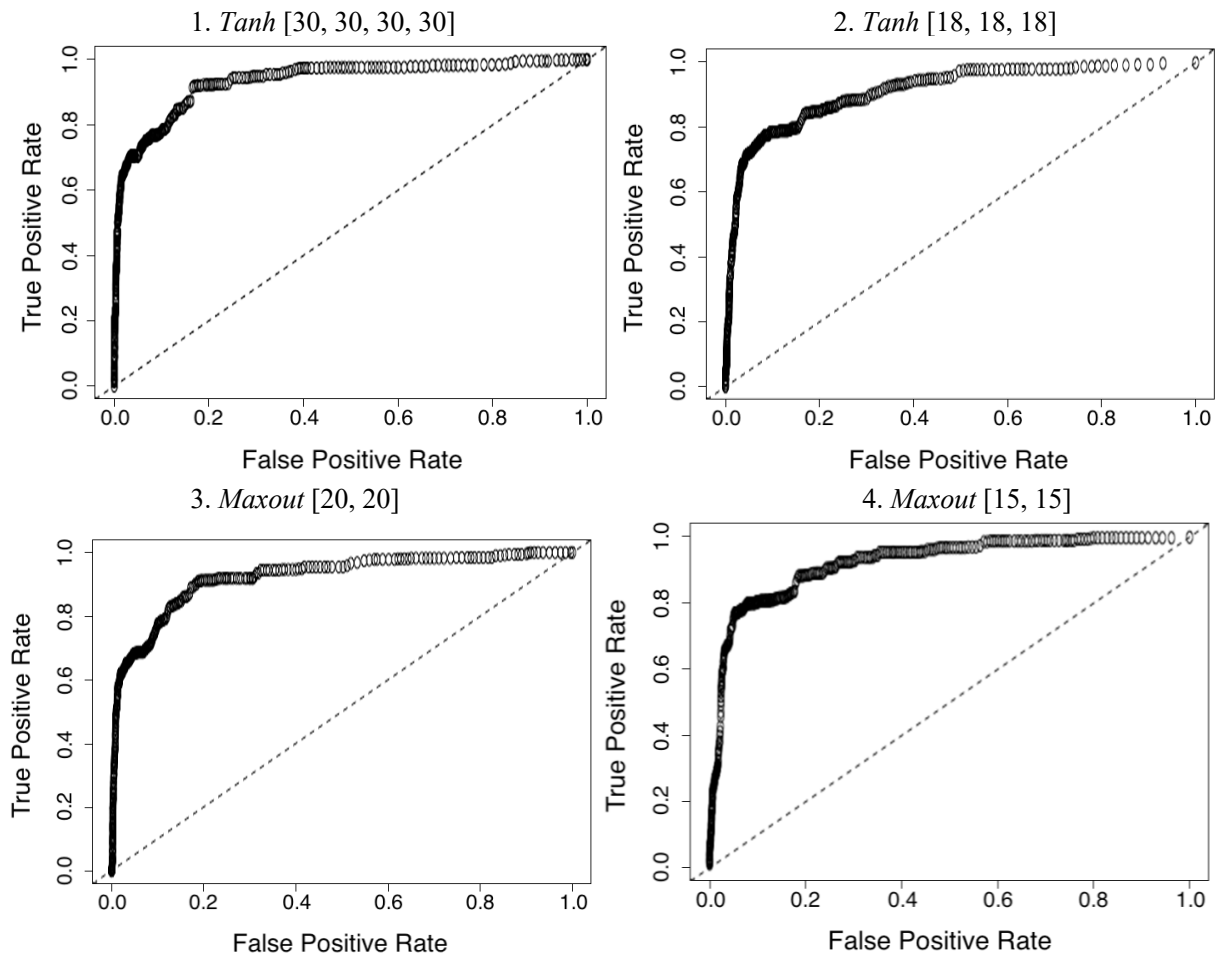


Fig. 6. Test Data ROC curves from DNN model

To further test the prediction accuracy of the DNN models, six false alarm rates (FAR) were determined for each model (3%, 5%, 7%, 10%, 20% and 30%) in order to determine the sensitivity of the model. Results show that Model 3 was the most sensitive model and consisted of 2 hidden layers, 20 nodes in each layer and the *maxout* as the activation function. In fact, it can predict 78% and 80% of the traffic conflicts at the cost of 5% and 10% false alarm rate respectively.

6. Discussion

From the 50 DNN models tested with a varying number of hidden layers, nodes, and two different activation functions, the best 4 models which achieved the highest AUC value with the testing data were presented. The model with the highest prediction for traffic conflicts consists of 4 hidden layers, 30 nodes in each layer and the *tanh* as the activation function. The model performance was also evaluated via 10-fold cross-validation. The results show that the AUC values obtained 0.935 on the testing data in this study. This exceeds that of previous work when using other machine learning classifiers which only achieved accuracies of up to 0.834 (Yang et al., 2017). This is because some of these previous studies employ machine learning predictors which cannot be used for highly imbalanced data sets. Therefore, some of their data may have been omitted which can affect the calibration of the model and therefore a high prediction accuracy is not achieved. As a result, by making use of the DNN network while considering a loss function which caters for highly imbalanced data sets (Wang et al., 2016), more accurate predictions are obtained. In this research the ‘Mean Squared False Error’ loss function was adopted because of its ability to capture classification errors from both the majority and minority classes equally and this shows in the prediction results obtained.

A real-time implementation of the model developed in this paper can be embedded to an ADAS system to warn drivers of any potential traffic conflicts. This is because once the DNN model is trained, there is no need to re-train it again. The real-time data is directly applied to the trained model in order to provide traffic conflict predictions. Moreover, if vehicles are connected with the ability to share information (e.g. platooning), this model can provide a coordinated traffic conflict prediction such that not only the ego-vehicle is controlled, but also the preceding vehicle. It is essential that the signals in proactive safety systems are released with enough time

1 for the driver to react and avoid these conflicts but not too hasty or too recurrent to become an
2 annoyance or interference to the driver. However, if the FAR is set too low so that the driver
3 does not become immune to the warnings, the prediction performance is very low. Nevertheless,
4 the prediction accuracy in this study from the ROC curves is relatively high and for an accepting
5 FAR of 5%, 71% of traffic conflicts can be predicted. If a threshold is selected to accept 10%
6 FAR, the prediction accuracy increases and around 78% of the traffic conflicts are identified.
7 This performance is sufficient for implementation in a real-time warning technology based on
8 the models presented in this paper (Yu and Abdel-Aty, 2014).

9 Moreover, these sensitivity results also show that these models are more sensitive than
10 previous studies which underline the identification ability in real-time traffic conflicts prediction
11 that the DNN technique has. However, the most sensitive model found from the 4 best models is
12 not the same one which obtained the highest AUC for the testing data. In fact, Model 3 was the
13 most sensitive model given that for a 10% FAR it is able to predict 83% of traffic conflicts.

14 While this process is time-consuming, this was only carried out to generate traffic
15 conflict data as an output variable to correspond to the input data from which the DNN model
16 can learn. It is important to note that once the output data is generated, the R-CNN model does
17 not need to be re-run again. However, if the precision of the R-CNN network needs to be further
18 improved, re-running of the model is required by modifying its parameters, adding more layers
19 and more training data on which the network. Moreover, a Faster R-CNN can be employed to
20 further enhance the knowledge gained from the images and obtain more reliable traffic conflicts
21 data and avoid any erroneous data.

22 Previous studies have confirmed that the real-time prediction models cannot be directly
23 transferred from one road to another because of the variations in traffic patterns and driver

1 population. Since this DNN model is based on motorway traffic readings, this model cannot be
2 used for an urban setting. However, transferability can be possible for roadways with similar
3 roads such as other motorways. Furthermore, the methodology can be applied to any type of road
4 network.

5 While the proposed DNN outperformed other machine learning classifiers, there are some
6 limitations in the study such as the lack of detail in the data namely the weather conditions,
7 drivers, road design which can improve the performance of the DNN. These added variables
8 could be beneficial for the model to improve the prediction accuracy further since they can
9 characterise a traffic condition. Moreover, in this work 50,000 data points were dealt with and by
10 increasing the size of the data set the predictability of these models is expected to increase.
11 Therefore, future work will include more testing especially in different weather conditions, to
12 further assess the performance of the model and tune it accordingly as well as testing the
13 predictive performance of the DNN model on a validation model.

14 **7. Conclusion**

15 This paper presented a new model for predicting traffic conflicts by using deep learning. A
16 centralised digital architecture was developed to integrate large volume and heterodox in-vehicle
17 sensors data with highly disaggregated traffic data. From this architecture, it was possible to
18 develop relationships between different surrogate safety measures (SSMs) derived from in-
19 vehicle sensors data and traffic variables such as speed, density and flow. For instance, time-to-
20 collision (TTC) – an SSM varies with the mean travelling speed indicating that a single threshold
21 value of TTC may not be appropriate for detecting a potential collision at different speeds. An
22 image processing technique aided by an R-CNN was employed to identify all the vehicles with
23 respect to the ego-vehicle from the video data, and several criteria were formulated to generate

1 the output variable for each input. The output variable was able to distinguish between traffic
2 conflicts and safe road dynamics which were subsequently integrated with the SSMs and the
3 traffic variables to develop the DNN model. From the developed DNN model continuously
4 running at 15Hz, real-time predictions of traffic conflicts were achieved because of its capability
5 to be applied on a highly imbalanced dataset.

6 Real-time traffic conflict prediction model offered a high prediction, accuracy, sensitivity
7 and a low false alarm rate. In fact, results show that the DNN network can predict 71% and 78%
8 of the traffic conflicts at the cost of 5% and 10% false alarm rate respectively.

9 Moreover, the results show that the best DNN model provides an accuracy of 94% and as
10 a result, have the potential to be used in ADAS to develop proactive safety management
11 strategies for improving traffic safety. Furthermore, if vehicles are connected, active safety
12 features (e.g. early prediction of traffic conflicts) can be shared between these vehicles and
13 therefore the risk of traffic collisions are mitigated.

References

- Abdel-Aty, M., Uddin, N., Pande, A., 2005. Split Models for Predicting Multivehicle Crashes During High-Speed and Low-Speed Operating Conditions on Freeways. *Transp. Res. Rec. J. Transp. Res. Board* 1908.
- Archer, J., 2005. Indicators for traffic safety assessment and prediction and their application in micro-simulation modelling: A study of urban and suburban intersections. Acad. thesis, R. Inst. Technol. Dep. Infrastructure, Div. Transp. Logist. Cent. Transp. Res. Royal Institute of Technology, Stockholm, Sweden. doi:ISBN 91-7323-119-3
- Bham, G., 2009. Estimating Driver Mandatory Lane Change Behavior on a Multi lane Freeway. *Transp. Res. Board 88th Annu. Meet.* 573 .
- Candel, A., Parmar, V., 2015. Deep Learning with H2O Deep Learning with H2O September .
- Chan, C.-Y., 2006. Defining Safety Performance Measures of Driver-Assistance Systems for Intersection Left-Turn Conflicts. *Intell. Veh. Symp. (IV)*, IEEE 25–30.
doi:10.1109/IVS.2006.1689600
- Chen, R., Kusano, K.D., Gabler, H.C., 2015. Driver behaviour during lane change from the 100-car naturalistic driving study. *Traffic Inj. Prev.* 16, 1–10.
- Gedeon, T.D., 1997. Data Mining of Inputs: Analysing Magnitude and Functional Measures. *Int. J. Neural Syst.* 08 02 , 209–218. doi:10.1142/s0129065797000227
- Guido, G., Saccomanno, F., 2010. Comparing safety performance measures obtained from video capture data. *J. Transp. Eng.* 137 July , 481–491. doi:10.1061/(ASCE)TE.1943-5436.0000230.

1 He, H., Garcia, E.A., 2009. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* 21 9
2 , 1263–1284. doi:10.1109/TKDE.2008.239

3 Hossain, M., Muromachi, Y., 2013. Understanding crash mechanism on urban expressways
4 using high-resolution traffic data. *Accid. Anal. Prev.* 57, 17–29.
5 doi:10.1016/j.aap.2013.03.024

6 Katrakazas, C., Quddus, M.A., Chen, W.-H., 2017. A new methodology for collision risk
7 assessment of autonomous vehicles. *Transp. Res. Board 96th Annu. Meet.*

8 Kusano, K.D., Montgomery, J., Gabler, H.C., 2014. Methodology for identifying car following
9 events from naturalistic data. *IEEE Intell. Veh. Symp. Proc. Iv* , 281–285.
10 doi:10.1109/IVS.2014.6856406

11 Laureshyn, A., Svensson, Å., Hydén, C., 2010. Evaluation of traffic safety, based on micro-level
12 behavioural data: Theoretical framework and first implementation. *Accid. Anal. Prev.* 42 6 ,
13 1637–1646. doi:10.1016/j.aap.2010.03.021

14 Li, X.R., Jilkov, V.P., 2003. Survey of Maneuvering Target Tracking. Part I: Dynamic Models.
15 *IEEE Trans. Aerosp. Electron. Syst.* 39 4 , 1333–1364. doi:10.1109/TAES.2003.1261132

16 Mohr, D., Muller, N., Krieg, A., Gao, P., Kaas, H.-W., Krieger, A., Hensley, R., 2013. The road
17 to 2020 and beyond: What’s driving the global automotive industry? McKinsey Co. 24.
18 doi:10.1007/978-3-319-02868-2_3

19 Nadimi, N., Behbahani, H., Shahbazi, H., 2016. Calibration and validation of a new time-based
20 surrogate safety measure using fuzzy inference system. *J. Traffic Transp. Eng. (English Ed.*
21 3 1 , 51–58. doi:10.1016/j.jtte.2015.09.004

- 1 Naito, A., Miyajima, C., Nishino, T., Kitaoka, N., Takeda, K., 2009. Driver evaluation based on
2 classification of rapid decelerating patterns. ICVES 2009 - 2009 IEEE Int. Conf. Veh.
3 Electron. Saf. 108–112. doi:10.1109/ICVES.2009.5400239
- 4 Najafabadi, M.M., Villanustre, F., Khoshgoftaar, T.M., Seliya, N., Wald, R., Muharemagic, E.,
5 2015. Deep learning applications and challenges in big data analytics. J. Big Data 2 1 , 1–
6 21. doi:10.1186/s40537-014-0007-7
- 7 NHTSA, 2015. Critical reasons for crashes investigated in the national motor vehicle crash
8 causation survey. (traffic safety facts crash•Stats). 13th Int. IEEE Conf. Intell. Transp. Syst.
9 2–3.
- 10 O'Reilly, U.-M., Yu, T., Riolo, R., Worzel, B., 2004. Genetic Programming Theory and Practice
11 II. Springer Science & Business Media.
- 12 Ozbay, K., Yang, H., Bartın, B., Mudigonda, S., 2008. Derivation and Validation of a New
13 Simulation-based Surrogate Safety Measure. Transp. Res. Rec. J. Transp. Res. Board 2083,
14 105–113.
- 15 Papazikou, E., Quddus, M., Thomas, P., 2017. Detecting Deviation from Normal Driving Using
16 SHRP 2 NDS Data. Proc. 96th Annu. Meet. Transp. Res. Board August 2016 , 0–15.
- 17 Papazikou, E., Quddus, M., Thomas, P., Kidd, D., 2019. What came before the crash? An
18 investigation through SHRP2 NDS data. Saf. Sci. 119 January 2018 , 150–161.
19 doi:10.1016/j.ssci.2019.03.010
- 20 Qu, Xiaobo, Kuang, Y., Oh, E., Jin, S., 2014. Safety Evaluation for Expressways: A
21 Comparative Study for Macroscopic and Microscopic Indicators. Traffic Inj. Prev. 15 1 ,

1 89–93. doi:10.1080/15389588.2013.782400

2 Qu, X, Yang, Y., Liu, Z., Jin, S., Weng, J., 2014. Potential crash risks of expressway on-ramps
3 and off-ramps: a case study in Beijing, China. *Saf. Sci.* 70, 58–62.

4 Rodríguez, S.A., Frémont, V., Bonnifait, P., Cherfaoui, V., 2010. Visual confirmation of mobile
5 objects tracked by a multi-layer lidar. *IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC*
6 849–854. doi:10.1109/ITSC.2010.5625200

7 Songchitruksa, P., Tarko, A., 2006. Practical method for estimating frequency of right-angle
8 collisions at traffic signals. *J. Transp. Res. Board* 89–97.

9 Tractica, 2016. Advanced Driver Assistance System Market Forecasts.

10 Vogel, K., 2003. A comparison of headway and time to collision as safety indicators. *Accid.*
11 *Anal. Prev.* 35 3 , 427–433. doi:10.1016/S0001-4575(02)00022-2

12 Wang, C., Quddus, M.A., Ison, S.G., 2013. The effect of traffic and road characteristics on road
13 safety: A review and future research direction. *Saf. Sci.* 57, 264–275.
14 doi:10.1016/j.ssci.2013.02.012

15 Wang, S., Liu, W., Wu, J., Cao, L., Meng, Q., Kennedy, P.J., 2016. Training Deep Neural
16 Networks on Imbalanced Data Sets. *2016 Int. Jt. Conf. Neural Networks* 4368–4374.
17 doi:10.1109/IJCNN.2016.7727770

18 Wang, T., Wang, C., Qian, Z., 2017. Development of a new conflict-based safety metric for
19 freeway exit ramps. *Adv. Mech. Eng.* 9 9 , 1–10. doi:10.1177/1687814017723286

20 Wang, X., Fan, T., Chen, M., Deng, B., Wu, B., Tremont, P., 2015. Safety modeling of urban

- arterials in Shanghai, China. *Accid. Anal. Prev.* 83, 57–66. doi:10.1016/j.aap.2015.07.004
- Xu, C., Wang, W., Liu, P., 2013a. A genetic programming model for real-time crash prediction on freeways. *IEEE Trans. Intell. Transp. Syst.* 14 2 , 574–586. doi:10.1109/TITS.2012.2226240
- Xu, C., Wang, W., Liu, P., 2013b. Identifying crash-prone traffic conditions under different weather on freeways. *J. Safety Res.* 46, 135–144. doi:10.1016/j.jsr.2013.04.007
- Yamashita, R., Nishio, M., Do, R.K.G., Togashi, K., 2018. Convolutional neural networks: an overview and application in radiology. *Insights Imaging* 9 4 , 611–629. doi:10.1007/s13244-018-0639-9
- Yang, K., Wang, X., Quddus, M.A., Yu, R., 2018. Deep learning for real-time crash prediction on urban expressways using highly imbalanced big data. *J. Transp. Res. Board.*
- Yang, K., Yu, R., Wang X, X., Quddus, M., 2017. Utilizing stochastic gradient descent for model updating in real-time crash risk prediction. *Transp. Res. Board 96th Annu. Meet.*
- Yang, M., Wang, X., Quddus, M., 2019. Examining lane change gap acceptance, duration and impact using naturalistic driving data. *Transp. Res. Part C Emerg. Technol.* 104 April , 317–331. doi:10.1016/j.trc.2019.05.024
- Yu, R., Abdel-Aty, M., 2014. An optimal variable speed limits system to ameliorate traffic safety risk. *Transp. Res. Part C Emerg. Technol.* 46, 235–246. doi:10.1016/j.trc.2014.05.016
- Yu, R., Abdel-Aty, M., 2013. Utilizing support vector machine in real-time crash risk evaluation. *Accid. Anal. Prev.* 51, 252–259. doi:10.1016/j.aap.2012.11.027

- 1 Zaki, M., Sayed, T., Tageldin, A., Hussein, M., 2013. Application of Computer Vision to
2 Diagnosis of Pedestrian Safety Issues. *Transp. Res. Rec.* 2393 , 75–84. doi:10.3141/2393-09
- 3 Zhao, Z.-Q., Zheng, P., Xu, S., Wu, X., 2018. Object Detection with Deep Learning: A Review
4 14 8 .
- 5 Zheng, B., Liang, H., Zhu, Q., Yu, H., Lin, C.W., 2016. Next generation automotive architecture
6 modeling and exploration for autonomous driving. *Proc. IEEE Comput. Soc. Annu. Symp.*
7 *VLSI, ISVLSI* 53–58. doi:10.1109/ISVLSI.2016.126
- 8 Zheng, L., Ismail, K., Meng, X., 2014. Traffic conflict techniques for road safety analysis: open
9 questions and some insights. *Can. J. Civ. Eng.* 41 7 , 633–641. doi:10.1139/cjce-2013-0558

10